

关于计算机科学与技术认知问题的 研究简报 (I, II)

赵致琢

(厦门大学计算机科学系 厦门 361005)

(zzzhao@xmu.edu.cn)

摘 要 计算机科学与技术的认知对学科的科学研究与人才培养具有重要的意义。对计算机科学与技术学科发展的主线、特点和内在规律进行了回顾与总结,从科学哲学的角度阐述了学科的一系列认知问题,并概要介绍了学科方法论研究的主要进展,讨论了学科方法论研究进展对学科未来发展和人才培养可能产生的影响。

关键词 学科方法论, 基本问题, 学科形态, 核心概念, 典型方法, 典型实例, 思维方式数字化

中图法分类号 TP301

THE BRIEF RESEARCH REPORT ON THE COGNITION OF COMPUTER SCIENCE AND TECHNOLOGY

ZHAO Zhi-Zhuo

(Department of Computer Science, Xiamen University, Xiamen 361005)

Abstract The cognition of computer science and technology is very important for studying computer science, and predicting the developing directions of computer science in future. In this paper, the development mainlines, characteristics and internal inherent law are reviewed and summarized, and the cognitive process for computer science and technology is explained from the perspective of scientific philosophy. A series of new concepts and idea, for example, the essential problem, typical method and typical instance, mathematicalization for mode of thinking, core-courses for graduates, and intensive model for studying computer science are proposed and discussed. The research progress of methodology for computer science and technology is outlined, and its influence on the research and education of computer and technology is discussed.

Key words methodology, essential problem, paradigms, core-concept, typical method, typical instance, mathematicalization for mode of thinking

1 引 言

作为 20 世纪新兴的技术学科,计算机科学与技术(以下简称计算科学)在短短的几十年里获得了空

前的发展,其研究和开发渗透到社会生活的各个方面,在大多数学科中获得了广泛的应用,影响和改变着人类的思维模式和行为方式,成为各国争相发展的重点科技领域。

正如现代科学技术的发展既高度分化,又高度

综合,出现加速发展的特点一样,计算科学在发展中出现了既不断深化,又与其它学科建立密切联系并发展具体应用的特点.学科知识体系日渐庞大,学科研究热点变换频繁,以至学术界在计算科学深度与广度的发展认识上整体把握未来的发展走向面临困难.今天,如何全面把握学科发展的特点、内在规律和未来发展的趋势,按照内涵发展的客观要求,建立学科发展的(应用)基础研究创新体系是一个受到广泛关注而又亟待解决的理论问题.

厦门大学计算机科学系从 1995 年起,结合教育部的有关项目,对计算科学的认知基础进行了深入的研究,提出了一系列学术新观点和新思想,并结合教育与教学改革目标建立了一个学科人才培养的科学理论体系(框架).本文是学科认知基础与上述科学理论体系(框架)研究成果核心内容与研究进展的简报,许多内容是第 1 次发表.

本文运用科学哲学的思想方法,从内涵发展方式上系统地总结了学科发展的历程,研究了若干学科认知问题和构成学科方法论的若干重要内容,并以此为基础,对学科未来的发展提出了一些新的思想和观点,有助于学术界把握学科未来发展和对学科的认知.

2 学科发展的主线、特点和内在规律

2.1 关于学科的定义

60 年代,计算科学被认为是算法的学问,学科的基本问题是算法问题.20 多年以后,随着学科研究与发展的深入,一批学者开始认识到原来的定义不准确,随后给出了新的学科定义^[1]:

计算机科学与技术是对描述和变换信息的算法过程,包括其理论、分析、设计、效率分析、实现和应用的系统的研究.全部计算机科学与技术学科的基本问题是:什么能(有效地)自动进行.

由于缺乏深入的理论分析,事实上,这个定义仍然不够准确和完整.近年来,我们通过系统分析和总结学科发展历程,在分析和提炼学科的基本问题、发展主线、学科特点和内在规律的基础上,给出了新的学科定义,并提供了详细的论证^[2~5].

计算机科学与技术是对描述和变换信息的算法过程,包括其理论、分析、设计、效率分析、实现和应用的系统的研究.全部计算机科学与技术学科的基本

本问题是:什么能(有效地)自动进行,什么不能(有效地)自动进行.本学科来源于对数理逻辑、计算模型、算法理论和自动计算机器的研究,形成于本世纪 30 年代后期.

2.2 学科的基本问题

任何一个学科在发展的历程中,总是围绕着解决学科的基本问题和重大问题不断地向前发展.那么,哪些问题是计算科学的基本问题和重大问题呢?

重大问题是比较容易理解的,而且,相对于不同时期,重大问题既是相对的,也是比较多的.例如,学科发展早期提出的什么是自动计算装置的理论模型,什么是可计算与不可计算的概念,50 年代末 60 年代初提出的高级程序设计语言的形式化描述问题,60 年代末 70 年代初提出的 NP 完全性问题,70 年代末 80 年代初提出的人工智能的逻辑基础问题,网络通信协议的描述问题,80 年代末 90 年代初提出的可变结构的计算模型问题以及当前的量子计算、生物计算机等.

在经历了几十年的发展之后,当我们今天以科学哲学的思想方法回顾学科的历史进程,系统总结学科的发展内容时,可以看到:如同数学等一些基础学科一样,在学科各个分支方向的发展进程中,不断地出现了一些在表现形式上虽然不同,但在科学哲学的解释下本质上是相同或相近的问题,即学科研究与发展普遍关心的基本问题.根据学科定义,学科的基本问题本质上是能行问题.但是,对学科基本问题的这样一种描述显然不够精细,难以把握.通常,能行性更多地被认为体现的是学科的基本特点.经过深入分析,我们认为,学科的基本问题主要可通过下列 3 个基本问题体现:

- (1) 计算的平台与环境问题;
- (2) 计算过程的能行操作与效率问题;
- (3) 计算的正确性问题.

历史上,为了实现自动计算,人们首先发明和制造了自动计算机器.在实际的探索过程中,前人逐渐认识到,研究自动计算不仅要理论上提供计算的平台——观察和描述计算的起点,用于解题或者证明问题本身不可解,而且要实际制造出针对各种待处理问题特点和要求的自动计算机器,这就产生了计算模型、计算机原理与计算机体系结构的研究.那么,是否所有的计算模型都是能行的呢?答案是否定的.众所周知,只有构造性计算模型才有可能能行的,这类计算模型所采用的构造性数学描述方法保证了计算的能行性.但是,并不是凡具有能行性特点

的计算模型都能够造出具体的自动计算装置,如非确定性计算模型。这说明,计算模型的研究从应用的角度考虑,必须注重区分计算模型在理论上的能行性和现实中的能行性。进一步,从广义计算的概念出发,计算的平台在使用上还必须比较方便,于是派生出计算环境的概念。据此,不难看出,研究中提出的各种计算模型、实际的计算机系统、高级程序设计语言、计算机体系结构、软件开发工具与环境、编译程序与操作系统等都是围绕解决这一基本问题发展而来的,都属于计算的平台与环境问题的研究,其内容实质可归结为各种计算模型的研究。换言之,这个基本问题实际上关心的是计算问题在理论上和/或现实中是否能行和使用方便的问题。

计算过程的能行操作与效率问题也是学科的基本问题之一。一个问题在判明为可计算的性质后,为了具体解决这个问题,必须按照能行可构造的特点与要求,给出实际解决该问题的一步一步的能行具体操作,同时还必须确保这样一种过程的开销成本是用户能够承受的。围绕这一问题,学科发展了大量与之相关的研究内容与分支学科方向。例如,数值与非数值计算方法、算法设计与算法分析、代数化简、结构化程序设计的效率分析、以计算机部件为背景的数字系统逻辑设计技术、密码学与快速算法、演化计算、程序设计方法学(程序设计的技术方法)、自动布线、自动推理技术等都是在围绕这一基本问题展开、发展而形成的。显然,计算操作的能行与效率问题也涉及到了计算模型问题,但是,这里所说的计算模型既包括类似于具有状态转换特征的那一类计算模型(数学机器),也包括数学建模这类计算模型。不难看出,这一基本问题的核心是基于某一恰当计算模型的算法问题。这也辅助说明了在学科发展的早期,当计算的平台与环境、计算的正确性问题尚不突出时,为什么学术界将本学科看成是算法的学问。

计算的正确性在任何计算工作都不能回避的问题,特别是使用自动计算机器进行的各种计算。一个计算问题在给出能行操作序列的同时,必须确保计算的正确性,否则,计算是无意义的。围绕这一基本问题,长期以来,学科发展了一些相关的研究内容与分支学科,例如,算法理论(数值与非数值算法设计的理论基础)、程序设计语言的语义学、程序理论(程序描述与验证的理论基础)、程序测试技术、电路测试技术、软件工程技术(形式化的软件开发方法学)、计算语言学、容错理论与技术、Petri网理论、CSP理论、CCS理论、进程代数与分布式事件代数、分布式

网络协议等都是针对为解决这一基本问题而发展形成的。由于数学为各种计算方法提供了计算正确性的理论基础,而各种计算在计算机系统上的自动进行均采用语言(包括电路)描述,以程序或电路系统的载体形式出现,因此,计算的正确性问题常常归结为各种语言的语法和语义问题,研究的方式方法一般为先发展某种合适的计算模型(如开关电路),用计算模型来描述各种语言的语法和语义。特别,由于描述和实现计算离不开语言(包括电路),而语言的词法、文法已经比较成熟,当前的难点往往在语义,因此,这也从一个侧面揭示了计算的正确性问题常可以归结为语言的语义学问题,揭示了语义学在整个学科中的重要地位。

应该指出,上述基本问题并非仅出现在某些分支学科方向,而是普遍出现在各个分支学科方向之中,并且常交织在一起,是学科研究与发展中经常面对而又必须解决的问题。学科的3个基本问题从本质上分析,又都具有能行特点,可以归结为能行问题。首先,计算的平台与环境问题可以归结为计算模型问题。根据计算概念的定义,计算模型必须能行。这样,第1个基本问题就常归结为计算模型的能行问题。其次,根据上述分析,计算过程的能行操作与效率问题可以归结为基于某一恰当计算模型的算法问题。按照算法的定义,算法问题的研究实际上分成两个方面,一个是算法的设计描述问题,另一个是算法的效率问题。由于所讨论的算法都是基于某种计算模型之上的算法,故计算模型的能行性决定了这个问题本质上可归结为一个理论上的能行过程描述问题和现实条件下的能行问题。第3,计算的正确性问题也可以归结为能行问题。与上述问题不同,前两个问题是从实际解决问题的角度出发,客观上要求计算模型与算法研究应该注重能行性,而计算的正确性研究客观上并不要求一定要满足能行性,只要能够判定计算是正确的即可。但是,由于计算的正确性问题是上述两个基本问题研究的基础上针对一类诸如程序、逻辑公式、电路系统等对象进行语义判定,而前述基本问题的能行性特点决定了从过程的角度采用外延和内涵的方法进行语义分析研究时,必然使计算的正确性研究具有能行性特点。而且,目前大多数计算语义学研究从将来语义判定自动化的角度考虑,先后引入了一批成熟的能行计算模型和/或发展了一批构造性数学方法,将语义描述与判定融入在描述计算的过程中。因此,迄今为止,大量关于语义学研究的成果都具有能行性的特点。所以,限

于目前学术界对计算的认识, 计算的正确性问题可以归结为能行问题。至于计算的正确性问题能否采用非能行的方法或非构造性数学方法进行研究, 目前尚未见报道。

至此, 可以看出, 原来学科定义的不足与不完整之处在于该定义仅注意了学科中可以计算的问题。从理论和应用的角度看, 以古典的二值逻辑作为学科的逻辑基础不足以准确刻画可计算与不可计算的边界, 不可计算的问题及其判定同样重要。随着学科发展的不断深化, 将会较多地出现一些本质上与某个不可计算问题等价的复杂问题。为了避免无谓的科技投入浪费大量人力物力, 充分认识可计算与不可计算的边界, 学科定义、研究与发展必须突出不可计算与非能行性的判定问题。

循着这一线索, 可以看出, 整个学科正是在围绕这些基本问题和一些重大问题而展开的研究与开发中逐步形成了学科发展的主线与主流方向。

2.3 学科发展的主线

任何一门学科都存在许多分支学科和研究方向, 然而, 在学科的发展过程中, 不同时期, 围绕着学科的基本问题和一些重大问题, 若干方向便构成了所谓的主流方向, 由主流方向又形成了学科发展的主线。

在计算科学发展的历程中, 不断地追求制造出各种新型计算机系统, 拓展和提高计算机的应用领域和应用水平这样两个目标使学科在发展中围绕基本问题和重大问题的研究, 逐步形成了 3 条相对独立的主线, 他们是: 计算模型与计算机, 计算模型、语言与软件开发方法学, 应用数学与计算机应用^[5, 6]。

2.4 学科发展的特点和内在规律

计算科学是在数学和电子科学基础上发展起来的一门新兴学科。它既是一门理论性很强的学科, 又是一门实践性很强的学科。几十年来学科自身发展的实践表明, 这门学科实际上是一门偏理的技术科学^[2~5]。

事实上, 理论和技术是学科两个互为依托的侧面, 由于能行性性质的作用, 学科理论与技术之间的高度一致透射出学科的理论绝大多数属于技术理论。学科的基本问题和本质属性决定了学科理论、技术与工程相互之间常常界限模糊。从理论探索、技术开发到工程开发应用和生产的周期很短, 许多实验室产品和最终投向市场的产品之间几乎没有太大的差别, 这在软硬件产品中是屡见不鲜的。虽然, 目前整体上理论研究滞后于技术开发, 但随着学科研究

和应用的不断深化, 理论的重要性地位将愈来愈突出, 而技术则渐渐退居为次要的位置。大量的研究已经表明, 像新一代计算机体系结构、软件工程、并行与分布式计算、计算语言学、容错计算、人工智能等许多方面的难题并不是技术问题, 而恰恰是理论问题。这些难题基本上都与某些数学理论或工具之间存在着密切的联系, 它们的解决将对学科的发展产生极其深远的影响。

数学是计算科学的主要基础, 以离散数学为代表的应用数学是描述学科理论、方法和技术的主要工具, 而微电子技术和程序技术则是反映学科产品的主要技术形式。在学科中, 无论是理论研究还是技术研究的成果, 最终目标要体现在计算机软硬件系统产品和技术服务上。然而, 作为硬件产品的计算机系统 and 作为软件产品的程序指令系统应能机械地, 严格地按照程序指令执行, 决不能无故出错。计算机系统的这一客观属性和特点决定了计算机的设计、制造, 以及各种软件系统开发的每一步都应该是严密的、精确无误的。就目前基于图灵机这一理论计算模型和存储程序思想设计制造的计算机系统而言, 它们只能处理离散问题或可用构造性方式描述的问题, 而且这些问题必须对给定的论域存在有穷表示。至于非离散的连续性问题, 如实数域上的函数计算, 方程求根等还只能用近似的逼近方法。于是, 由于计算模型的非连续性(或称离散化)特点, 使得以严密、精确著称的数学尤其是离散数学被首选作为描述学科的主要工具。

在计算科学发展中, 不仅许多理论是用数学描述的, 而且许多技术也是用数学描述的。大多数计算科学理论不仅仅是对研究对象变化规律的陈述, 而且由于能行性这一本质的核心问题和特点的作用, 理论描述中常通过构造性方法折射出技术的思想和步骤, 而从理论通过方法跨越到技术则完全取决于对理论的深刻认识和理解。由于离散数学的构造性特征与反映学科本质的能行性特征之间形成了天然一致, 从而使离散数学的构造性特征决定了学科的许多理论同时具有理论、技术、工程等多重属性, 决定了其许多理论、技术和工程的内容常常是相互渗透在一起的, 是密不可分的。与大多数工程科学的工作方式不同, 在几乎所有高起点的、有学术深度的学科研究与开发中, 企图参照经验科学的工作方式, 通过反复实验获得数据, 经分析后指导下一步的工作从而推进科研与开发工作的方式是行不通的。原因是有学术深度的问题其复杂性早已大大超出了专家

们的直觉和经验所能及的范围

不难看出, 在学科 3 条主线各方向上的研究工作都与应用数学有着密切的联系。理论上, 凡是可以计算机来处理的问题及其处理过程都可以用应用数学来描述; 凡是可以用以离散数学为代表的构造性数学描述的问题及其处理过程, 只要所涉及的论域是有穷的, 或虽为无穷但存在有穷表示, 也一定可以用计算机来实现。至于现实是否能行, 则主要取决于计算的复杂性。所谓实现, 是指以某种技术形式来表示和反映计算模型与计算过程。技术形式可以是程序技术, 也可以是微电子技术, 甚至还可以是光电子技术或生物技术。但不管是什么技术形式, 计算模型与计算过程的“严密性”和“精确性”都决定了所采用的技术与理论之间, 特别是技术相对于理论而言偏差是极小的, 否则, 这种技术就不适合作为计算科学中的实现技术。

计算科学并不完全排斥工程的方法。相反, 学科在发展中广泛采用了其它学科行之有效的工程方法。如在软件开发中认识到并采用首先开发工具和环境, 进而开发软件的方式方法; 在计算机的设计中, 目前广泛使用标准组件的方法; 在软件的设计和质量检查中广泛使用程序设计规范、软件测试技术、标准化技术等。事实上, 从软硬件开发的经济效益和专业技术人才的局限性两方面考虑, 完全采用形式化的方法来解决学科中的所有问题是不现实的, 也是不必要的。

长期以来, 数学, 特别是以离散数学为代表的应用数学与计算科学之间建立的密切联系不仅使学科具备了坚实的理论基础和科学基础, 而且使学科从一开始就以一种与其它学科发展方式很不相同的方式发展, 其显著特点是在研究与开发中, 人们总是将抽象描述求解问题与具体实现解决问题相分离, 即对各种各样有深度的问题的处理采用了将理论上解决问题的抽象描述计算方法、算法和技术的内容与具体解决问题的细节、具体实现计算的技术内容相分离。

抽象描述与具体实现相分离是计算科学发展过程中一个十分重要的特点, 它不仅决定了一大批学科专业工作者的工作方式, 而且使学科的研究与开发在很短的几十年里就借助数学工具进入学科比较深层的研究阶段。特别在将抽象描述与具体实现相分离的过程中, 由于抽象描述与深入研究的需要, 计算模型的基础地位突出地表现出来, 并经过对各种计算模型的深入研究和应用使学科迅速走向深入。

由此可见计算模型在整个学科发展中起着不可替代的独特作用, 并产生了学科发展的另一个特点。

计算科学发展的另一个重要的特点是几乎在学科各个方向和各个层面, 一旦研究工作走向深入, 研究内容具有比较复杂的特点, 人们首先是发展相应的计算模型或引入新的数学工具, 然后依靠计算模型和数学工具将研究工作推向深入。例如, 网络协议描述、各种程序设计语言的语义描述、数据库理论、并发程序的语义描述、并发控制机制、容错计算、量子计算、计算机系统结构与性能分析、并行计算、自动推理技术、人工智能逻辑基础、VLSI 设计的数学理论等都引入了新的计算模型。

2.5 数学在整个学科发展中的地位与作用

数理逻辑是数学的一个分支, 是研究推理的科学。本世纪初, 关于数学基本问题的讨论产生了数理逻辑的三大学派, 即逻辑学派、形式主义学派和直觉主义逻辑学派, 极大地促进了构造性数学的发展, 并出现了具有相当强的计算能力的图灵机等一批计算模型, 在数理逻辑与计算模型之间建立了密切联系。数理逻辑之所以重要, 不仅在于它揭示了计算的极限, 而且成为学科大多数重要分支方向共同的基础。

数理逻辑与抽象代数是学科最重要的两项数学基础, 它们的研究思想和方法在学科许多有深度的领域得到了最广泛的应用。在今天, 世界上还没有一门科学像计算科学那样与数学的联系如此紧密, 因为数理逻辑与抽象代数在计算科学中最广泛的应用不仅仅解决了大量的学科问题, 而且它们反映和融入了现代数学研究的主流。

从科学哲学的角度进行总结, 可以从下列几个方面认识数理逻辑和代数是学科最主要的基础。

(1) 从计算模型和可计算性的角度看, 可计算函数和可计算谓词是等价的, 相互之间可以转化。这就是说, 计算可以用函数演算来表达, 也可以用逻辑系统来表达。作为计算模型可以计算的函数恰好与可计算谓词是等价的, 而且, 数理逻辑本身的研究也广泛使用代数方法, 同时, 逻辑系统又能通过自身的无矛盾性保证这样一种计算模型是合理的。由此可见, 作为一种数学形式系统, 图灵机及其与它等价的计算模型的逻辑基础是坚实的。

(2) 在实际计算机的设计与制造中, 使用数字逻辑技术实现计算机的各种运算的理论基础是代数和布尔代数。布尔代数只是在形式演算方面使用了代数的方法, 其内容的实质仍然是逻辑。依靠代数操作实现的指令系统具有(原始)递归性, 而数字逻辑

技术和集成电路技术只是计算机系统的一种产品的技术形式;

(3) 从计算机程序设计语言方面考察, 语言的理论基础是形式语言、自动机和形式语义学, 而形式语言、自动机和形式语义学所采用的主要研究思想和方法来源于数理逻辑和代数。一阶逻辑系统与数字逻辑电路本质上都是语言, 都具有一般语言所具有的共性, 也可以视为一种计算模型。语言可视为某种计算模型的外在表现形式, 语言属于计算模型概念的外延。程序设计语言中的许多机制和方法, 如子程序调用中的参数代换、传递和赋值, 标识符的作用域, 数字逻辑电路的设计原理和方法等都出自数理逻辑的内容和方法。此外, 在语言语义的研究中, 操作语义、指称语义、公理语义、代数语义等 4 种语义方法最终可归结为代数和逻辑的方法。这就是说, 数理逻辑和代数为语言学研究提供了方法论的基础。狭义的计算模型主要是指具有状态转换特征, 能够对被处理的对象进行表示、加工的数学系统, 而广义的计算模型应包括各种主要的计算平台;

(4) 在计算机体系结构的研究中, 像容错计算机系统、Transputer 计算机、阵列式向量计算机、可变结构的计算机系统结构及其计算模型等都直接或间接同逻辑与代数密不可分。如容错计算机的重要基础之一是多值逻辑, Transputer 计算机的理论基础是 CSP 理论, 阵列式向量计算机必须以向量运算为基础, 可变结构的计算机系统及其计算模型主要采用逻辑与代数的方法;

(5) 从计算机各种应用的程序设计方面考察, 任何一个可在存储程序式电子数字计算机上运行的程序, 其对应的计算方法首先都必须是构造性的, 数据表示必须离散化, 计算操作必须使用逻辑或代数的方法进行, 这些都应体现在计算方法、算法和程序之中。此外, 到现在为止, 程序的语义及其正确性的理论基础仍然是数理逻辑或进一步的模型论。

尽管计算科学是在数学与电子科学的基础上发展起来的一门技术科学, 但是, 从学科特点和学科方法论的角度考察, 学科的主要基础是数学, 特别是数学中以代数、逻辑为代表的离散数学, 而程序技术和电子技术仅仅只是学科产品的主要技术形式。一旦某种材料和技术能够制造出可表示二进制的两个稳定状态的器件, 那么, 电子技术完全可能被新技术取代。例如, 生物芯片技术、光电子材料技术都是很有希望的替代技术。

在计算机应用技术研究中, 结合具体领域, 也产

生了一批共性技术, 而且, 许多共性技术的研究也运用了传统的分析数学的内容。例如, 在计算机图形学、图像处理、信号处理、电路分析与模拟、计算机仿真等领域中, 解析几何、数学分析、微分几何、微分方程、复变函数、小波理论等得到广泛应用。然而, 应该看到, 在这些领域, 一旦要将各种数学应用付诸实现, 理论上的计算方法必须离散化, 必须运用数值分析中的近似方法, 如迭代法、逼近法等。可见, 以离散数学为代表的构造性数学依然是计算科学最主要的数学基础。

2.6 计算科学知识组织结构及其演变

计算科学的发展和重大突破离不开学科核心知识组织结构中各分支学科的发展和支撑。学科核心知识组织结构是从事学科高水平研究与开发人员所必需的专业基础知识。从知识组织的层次结构和方法论的角度观察, 分析目前学科的整体发展情况, 可以得出, 新型计算模型与体系结构、软件开发方法学与计算机应用技术是学科未来发展的主要方向, 而计算理论(包括算法理论)、高等计算机体系结构、高等逻辑、并行算法设计基础或分布式算法设计基础、形式语义学是支撑学科未来主要方向发展的核心专业知识基础。

如同 50 年代数学研究所需要的核心专业基础知识从高等代数、数学分析(含几何、分析、函数论)与微分方程向抽象代数、泛函分析与拓扑学过渡转移发生变化一样, 计算科学未来发展所需要的核心专业基础也正在发生转移, 这一客观规律和学科发展的趋势应该引起学术界严肃认真的对待。

3 学科方法论研究的意义与研究进展

着眼于解决计算机科学教育问题这一具有战略性的研究问题, 1991 年, IEEE/CS 与 ACM 联合攻关小组提出了一个 91 教学计划及其相关研究报告^[1, 6, 7](以下简称 91 计划与报告)。他们的工作发表后, 引起国内学术界的广泛重视, 同时也产生了不少疑问, 特别是一些认识问题和该计划如何实施的问题。应该承认, 尽管研究成果比较初步, 但 91 计划与报告在引导学科专业教育走向内涵发展模式方面具有开创性。^[2, 5]

大多数学科在发展的早期, 一般由各种问题引发了大量的研究工作并积淀了丰富的学科知识。随着研究内容、范畴的日渐明朗和研究成果的不断增加, 逐步经后人系统整理而开始形成新的学科。但

是, 这样一种学科发展的模式基本上是一种外延发展的主导模式, 不利于学科知识体系的总结和继承, 也不利于从整体上把握学科发展的内在规律和未来的发展趋势。于是, 如何将学科由外延发展模式转向内涵发展模式, 就成为一个重要的学科理论问题。为着眼于学科教学改革, 91 教学计划及其相关研究报告在这方面进行了初步的探索, 其研究工作具有某种里程碑意义。

应该指出, 91 计划与报告并没有意识到内涵发展模式的重要性。从 IEEE/CS 与 ACM 联合攻关小组最近在网上发表的总体报告不难看出, 秉承 91 报告的主要思想, 联合攻关小组的主要想法是通过界定学科范畴和主科目、学科知识点, 然后通过 3 个过程(学科形态)和核心概念将各种知识串联起来, 便于融会贯通。可惜, 他们的工作直到今天依然采用的是外延的研究方法^[8]。

91 计划与报告给出的 9 个学科分类科目基本上覆盖了学科的主要领域, 可是, 引入理论、抽象、设计 3 个学科形态或过程, 将学科形态与学科分类科目以二维矩阵的形式定义学科的研究范畴, 这种外延描述方式非但不能准确概括整个学科, 而且使人难以把握, 并容易误导人才培养方案走入外延模式。91 计划与报告希望用学科形态与核心概念来统领学科的教学, 使学科教学走向科学发展模式, 便于整体把握。但由于相关基础研究薄弱, 对学科的基本问题缺乏深入认识, 没有注意到重大问题、基本问题与学科发展的主线, 没有注意到学科知识组织的层次结构, 内涵的逻辑结构, 没有注意到学科方法论中典型方法与典型实例在内涵发展模式中的重要作用, 更没有注意到思维方式数学化的重要意义和由此在教学过程中强化了时间与空间制约因素的地位, 没有解决计算机科学理论与实验在整个学科中的地位与作用的认识问题, 因而没有解决如何从内涵角度把握整个学科及其发展的问题, 所以, 91 计划与报告的可操作性问题并没有解决。显然, 沿着分析学科发展的主线与主流方向、基本工作流程方式、学科形态与核心概念、典型方法与典型实例的研究路线, 弄清学科的基本问题和知识组织的层次结构, 弄清基础数学与离散数学、计算机科学理论、计算机技术理论、计算机科学实验在学科人才培养过程中的地位和作用, 有助于从科学哲学的角度从整体上把握学科发展的走向, 指导科学研究, 有助于用学科方法论从整体上把握学科教学的主要内容和重点, 提出学科人才培养的内涵发展模式, 也有利于培养

创新人才。

3.1 学科形态与核心概念

每一个学科都有其自身的知识组织结构, 学科形态、核心概念和基本工作流程方式。理论、抽象和设计是计算科学 3 个基本的学科形态^[1~3]。所谓学科形态是指从事该领域工作的文化方式。对这 3 种学科形态, 我们作如下解释:

第 1 种形态是理论, 基于学科的数学基础和理论, 广泛采用数学的研究方法。按照统一的合理的理论发展过程, 包含以下 4 个步骤:

- (1) 对研究对象的概念抽象(定义);
- (2) 假设对象的基本性质和对象之间可能存在的关系(定理);
- (3) 确定这些性质和关系是否正确(证明);
- (4) 解释结果(与计算机系统或研究对象形成对应)。

这个学科形态的基本特征是其研究内容的构造性数学特征, 是区别于更广泛的数学科学学科形态的典型特征。

第 2 种形态是抽象, 或称模型化, 基于学科的实验科学方法广泛采用实验物理学的研究方法。按照对客观现象和规律的实验研究过程, 包含以下 4 个步骤:

- (1) 确定可能世界(环境)并形成假设;
- (2) 构造模型并做出预言;
- (3) 设计实验并收集数据;
- (4) 分析结果。

这个学科形态主要出现在计算科学中与硬件设计和实验有关的研究之中。当学科理论比较深奥, 理解较为困难时, 不少科研人员在大致了解理论、方法和技术的条件下, 基于经验和技能常以这种学科形态方式开展工作。

第 3 种形态是设计, 基于工程, 广泛采用工程科学(如建筑工程)的研究方法。按照为解决某一个问题的构造系统或装置的过程, 包含以下 4 个步骤:

- (1) 叙述要求;
- (2) 给定技术条件;
- (3) 设计并实现该系统或装置;
- (4) 测试和分析该系统。

这个学科形态广泛出现在学科中与硬件、软件、应用有关的设计和实现之中。当学科理论(包括技术理论)已解决某一问题后, 科研人员在正确理解理论、方法和技术的条件下, 可以十分有效地以这种学科形态方式开展工作。

第 2 种学科形态抽象, 是对客观现象和规律的抽象, 是对客观现象和规律的描述和刻画. 在这一过程中, 与具体客观现象和规律无关或对具体客观现象和规律影响不大的因素被去掉了, 其内容构成了计算科学理论的一部分, 即技术理论中其内容比较难于用简洁的数学形式描述的那一部分. 第 1 种学科形态理论, 也是一种抽象, 是对客观现象和规律的数学化、形式化描述, 是对能行问题及其求解过程的刻画, 其内容构成了学科的数学基础和理论计算机科学. 本质上, 这两种学科形态都是对客观现象和规律的抽象, 只是方式的不同. 第 3 种学科形态设计是对学科理论的工程化实现.

理论和抽象在本质上的统一, 学科理论、技术理论与工程(实现)技术的高度一致性, 学科理论的构造性特征与全部学科的基本问题一起决定了计算机科学和计算机工程在本质上并没有区别.

在学科的研究与开发中, 3 种学科形态并不是孤立出现的, 它们常常交织在一起. 经验告诉我们, 没有抽象形态的支持, 理论研究就失去了背景参照; 没有理论的指导, 工程开发往往会误入歧途. 有许多例子分别表明, 抽象和设计阶段出现了理论; 理论和设计阶段需要模型化; 而理论和抽象阶段离不开设计, 它们必须考虑到现实是否能行.

3 种不同的学科形态实际上反映了在学科领域内从事工作的 3 种文化方式. 理论关心的是以形式化方式揭示研究对象的性质和相互之间的关系, 这是一种按照某种科学规律构筑人工科学的典型方式. 抽象关心的是以实验方式揭示研究对象的性质和相互之间的关系. 而设计关心的是以生产方式对这些性质和关系的一些特定的实现, 完成具体而有用的任务.

在学科的发展中, 有一批在各个分支学科中重复出现的概念. 它们虽然在各学科中的具体解释在形式上有差异, 但相互之间存在着重要的联系. 核心概念是学科重要思想、原则、方法、技术过程的集中体现, 有助于在学科的深层统一认识学科. 对核心概念的深入理解、正确拓展与应用的能力, 是计算科学家和工程师成熟的标志之一.

核心概念是方法论的重要组成部分, 一般具有如下特点:

(1) 在本学科的不少分支学科中经常出现, 甚至在学科中普遍出现;

(2) 在学科理论、抽象和设计这 3 个过程的各个层面上都有许多示例;

(3) 在理论上具有可延展和变形的作用, 在技术上具有高度的独立性.

下面, 我们把学科的核心概念作一个筛选和分类, 这样能更好地便于读者理解.

(1) 计算模型与能行性

计算模型 (computational model), 可计算性 (computability), 计算复杂性 (computational complexity), 最优性 (optimum), 相似性与对偶性 (similarity and duality);

(2) 抽象与构造性描述

论域与计算对象 (domain and computing object), 枚举 (enumeration) 与有穷表示 (finite representation), 分层与抽象的级 (hierarchy and levels of abstraction), 内涵与外延 (intension and extension), 递归 (recursion), 归纳 (induction), 自由与约束 (freedom and restriction);

(3) 系统特征

相容性 (consistency), 完备性 (completeness), 单调性 (monotonicity), 透明性 (transparency), 容错与安全性 (fault-tolerant and security), 开放性 (openness), 稳定性 (stability), 健壮性 (robustness);

(4) 计算方法

折衷 (compromise), 分解 (decomposition), 集成 (integration), 类比 (analogy), 推导 (inference or reasoning), 变换 (transformation), 扩展 (extension and expansion);

(5) 实现技术

类型 (type), 进程与线程 (process and thread), 顺序与并发 (sequence and concurrent), 软计算结点 (actor), 关联 (binding) 与实例化 (instantiation), 现役 (的) (active), 虚拟 (的) (virtual), 编码 (coding), 模式匹配 (pattern matching), 分权 (branching), 合一 (或通代) (unification), 循环与迭代 (loop and iteration), 重用 (reuse), 协议 (protocol), 规范与标准化 (standardization).

3.2 学科基本工作流程方式、典型方法与典型实例

对现实世界中被研究的对象进行抽象, 建立必要的基本概念, 运用数学工具和方法对其进行基础和应用基础研究, 研究(对象的)概念的基本性质、概念与概念之间的关系, 揭示对象发展变化的内在规律, 为实验设计和工程设计实现提供方法和技术, 并开展实验、工程设计与实现工作是学科的基本工作流程方式. 其中, 抽象(主要指抽象化过程)是理论、抽象和设计 3 个基本学科形态中最重要的一个形

态,它是连接学科科学研究与工程应用开发研究的重要环节^[2~5]。

根据学科知识组织的层次结构和学科发展的3条主线,在将学科形态贯穿在整个学科认知过程的始终,用核心概念将各分支学科的内在联系串联起来时,事实上,学科形态与核心概念并不能完全解决问题,还需要引入在这门学科中反复出现的具有哲学意义的典型方法,并通过一批典型实例给人以较深刻的感性认识,从而使对学科的认知过程转入内涵发展模式^[2~5]。

典型方法由一类具体的计算方法经方法论意义上的科学的抽象得到。典型方法是属于方法论的内容,一般具有如下特点:

(1) 在本学科的不少分支学科中经常出现,甚至在学科中普遍出现;

(2) 在学科理论、抽象和设计这3个过程的各个层面上都有许多示例,具有示范作用;

(3) 在理论上具有可延展和变形的作用,在技术上具有高度的独立性。

所谓典型实例是指那些反映学科某一方面内在规律和典型问题本质内容的实例。典型实例也是方法论的重要组成部分,由于它们往往以简化形式深入浅出地表达学科深奥的科学规律和学科典型问题的本质内容,因此,在学科研究中常常被用来辅助说明思想、原理、方法和技术,用来比较思想、理论、方法和技术的优劣。显然,典型实例在计算科学发展中具有重要的作用。典型实例一般具有如下特点:

(1) 在本学科的某些分支学科中出现,甚至在一些分支学科中普遍出现;

(2) 在理论上具有高度的抽象示例作用,具有深刻揭示理论内涵的作用,在技术上具有反映理论方法和技术方法的本质的作用,具有辅助验证和比较思想、系统、理论、方法和技术性能优劣的作用,在技术上具有高度的独立性。

就目前我们对学科的认识,归纳整理了下列6种典型方法和一大批典型实例。它们是内涵与外延的方法,以递归、归纳和迭代技术形式为代表的构造性方法、公理化方法、快速原型方法、演化方法以及展开与规约方法。

内涵与外延是哲学的两个基本的概念。所谓内涵是指一个概念所指称的事物的本质属性的总和,也就是概念的内容。外延是指概念所界定的所有对象的集合,即所有满足概念定义属性的对象的集合。内涵与外延的方法广泛出现在学科的许多分支中,

是一个能够对无穷对象的集合作分类处理的方法。

构造性方法是整个计算机科学与技术学科最本质的方法。这是一种能够对论域为无穷的客观事物按其有限构造特征进行处理的方法。这一方法在不同的分支学科的运用中具体形式可能不同,但其本质在哲学意义上是一致的。构造性方法以递归、归纳和迭代技术形式为其代表形式之一。除了众所周知的递归函数论(或可计算性理论)中使用递归定义和归纳证明技术,方程求根和函数计算中使用迭代技术外,在程序设计语言的文法定义和自然演绎逻辑系统的构造中,在关系数据理论模型和对象模型的研究中,以及在编译方法、软件工程、计算机原理、算法设计和程序设计中均大量使用了递归、归纳和迭代等构造性方法。例如,考察自然演绎的逻辑系统、程序设计语言和自然语言可知,它们都是一种语言。既然是一种语言,那么,语言的词法、语法、语义就成为研究的内容。在对这些内容进行研究的过程中,虽然在这3个方向上创造了形式上不同的概念和方法,也有相同的概念和方法,但本质上这些概念和方法在方法论上是相同的或相近的。如它们都具有构造语言成份的形式规则,进行演算或推导的运算规则,以及语义解释的定值方法,而这些形成规则、运算规则和语义解释的定值规则均可用构造性数学表述。构造性方法为计算科学中大量问题的能行性研究奠定了坚实的科学与技术方法论的基础。

公理化方法也是学科的一种典型方法。它能帮助学生认识一个系统如何严格表述,认识到完备性和无矛盾性对一个公理系统的重要性,认识每一条公理深刻的背景,独立性和它的作用。可惜,其深刻的哲学意义、学术深度和理解上的困难使得它很难为一般读者所熟练应用。将公理化方法熟练地运用于计算科学问题的研究是一个计算机科学家具有较高数学修养的重要标志之一。

在学科的基础和应用基础研究中,上述3种典型方法是比较常见的学科研究方法。下面,我们介绍在学科应用研究的工程技术开发研究中使用较多的另外3种典型方法。当然,某种典型方法适用于某一类研究并不等于不适用于其它类的研究,典型方法在学科的各个层面都具有广泛的例示,只是在不同的类别中运用时有的多一些,有的少一些。

快速原型方法是学科的典型方法。最初,快速原型的思想出现在软件工程的研究之中,其主要内涵是在软件的开发中,随着程序代码量的日渐庞大,开发费用和周期的不断增长,人们迫切需要对软件开

发中引入的新思想、新原理、新方法、新技术的可行性进行原理性验证,通过验证过程提出改进意见,为实际产品的工程技术开发提供原理性的指导。此时,Prolog 语言刚刚出现,研究人员很快发现,该语言特别适合充当快速原型实验的验证工具。对于软件开发中系统设计的一种新的构思,采用 Prolog 语言只对涉及新构思的内容“去粗取精”地开发原理性的验证程序,在证实了新构思确实可行之后,再按照规范的、工程化的要求进行软件的开发,并且在系统的开发中,基本上按照快速原型验证过程中的程序设计思想进行细化。这样做,不仅可以避免大型软件系统的开发出现原理性的错误,而且能够提高大型软件开发的 speed 和质量。显然,快速原型方法是一种低成本原理性验证方法。

演化方法也是学科的典型方法。所谓演化 (evolution) 方法,在一些文献上也叫进化方法。演化方法的主要思想是,针对具体的问题,首先找到解决该问题的办法 (或算法、程序、电路等),然后通过各种有效的技术方法改进解决问题的办法 (或算法、程序、电路等)。近年来在演化算法基础上发展起来的演化计算是该方法最具代表性的研究领域之一。例如,算法设计与分析中的代数化简方法,程序设计方法学中的程序变换方法,电路设计与分析中首先通过对电路对应的布尔函数进行化简,然后简化电路设计的方法等都是演化方法的具体应用实例。

另一种典型方法是所谓的展开与归约方法。展开与归约是一对技术概念,是在处理实际事务的过程中对两个相向的处理活动所作的一般化的方法学概括。展开的内涵是从一个较为抽象的目标 (对象) 出发,通过一系列的过程操作或变换,将抽象的目标 (对象) 转换为具体的细节描述。例如,在程序设计方法学中,从一个程序的规范出发,运用程序推导技术,可以将一个程序一步一步地自动设计出来,这样就实现了对该程序的展开。理论上,一个具体的程序与该程序的规范指称的是同一个对象,语义相同,只是表现形式不同。从描述与理解的角度看,前者较为具体,后者较为抽象。规约方法可以视为展开过程的逆过程。

在学科的研究与开发中,典型方法有时并不总是孤立出现的,它们常常交织在一起。长期的实践已经证明,许多复杂的计算科学问题的解决,仅仅使用一种典型方法是不够的,特别是在计算密集性和数据密集性的计算科学问题的求解中。

典型方法之所以重要,不仅在于它能深刻地揭

示学科的基本规律和各主科目之间内在的联系,而且在于它能启发人们更深入地思考学科的一些基本问题,有助于培养研究者的创造性能力和科学研究能力,有助于培养研究者正确的思想方法和技术开发能力。

当然,学科中还有许多方法也是重要的,如类比推理、证伪发现、统计推断与分析、各种测试方法与测试数据筛选的方法等。但相对于上述 6 种方法来说,这些方法应用范围比较单一、狭窄,或是哲学意义比较浅显。至于其它的典型方法,还有待人们去整理和总结。

在学科的发展中,经过几十年的研究与积累,人们构思和设计了一大批能够反映各分支学科有代表性的、具有问题本质特性的典型实例。下面是一组典型实例。

停机问题、哲学家共餐问题、最小费用流问题、可满足性问题、货郎担问题、生产与消费问题、文法的二义性问题、上下文语义问题、最小生成树问题、阿克曼函数、子集和数问题、3 个中国人算法、证比求易算法、0/1 背包问题、作业调度问题、路由选择问题、图的最小覆盖问题等。

4 学科方法论研究的进展对学科未来发展和人才培养的影响

学科方法论的研究具有重要的意义,其成果不仅有助于学科科学研究和人才培养走向内涵发展模式,而且对学科的深入发展具有一定的指导作用。

4.1 对学科未来发展的影响

学科方法论研究的进展必将对学科未来的发展和人才培养产生重大影响。一旦弄清了学科的定义、学科的基本问题、学科发展主线与主流方向、学科形态与核心概念、学科基本工作流程方式、典型方法与典型实例、学科发展特点和内在规律,人们就可以从科学哲学的角度和高度,观察和把握整个学科未来发展的走向,就比较容易预见学科未来的发展趋势。下面,我们依据学科发展的主线,针对学科的基本问题和一些重要问题,就典型方法对学科发展的影响进行讨论。

(1) 计算模型

在计算模型的研究中,早期主要是研究具有状态转换特征的计算模型或数学机器,产生了存储程序式计算机原理。在存储程序式计算机出现以后,以

制造新型计算机系统为背景的计算模型的研究主要集中在计算机逻辑结构优化的研究, 产生了 RISC 技术, 具有多功能部件的高性能计算机系统 1980 年之后, 人工智能计算机系统的研究开始集中在各种非经典逻辑的研究方面, 试图在存储程序式计算机的基础上通过增加各种逻辑系统平台开发智能计算机系统 总的来看, 因为各种新型计算模型的探索研究始终没能超越图灵机的计算能力, 跳出图灵机所确定的范畴, 计算模型的研究层面在不断上升, 研究内容逐渐远离可计算概念的实质

虽然对定义在实数集合之上的 BSS 机器的研究已有一些成果发表, 但关于计算模型方面的研究工作的主流仍然集中在侧重结构、内部控制机制的并行与分布式计算模型的研究方面, 集中在能够刻画描述连续现象语言语义的计算模型方面, 特别是计算模型内部成分和结构可以动态改变与变形的研究有可能是一个非常具有前途的方向 因为, 不仅研究所使用的模型论方法学术层次较高, 而且与生物内部微观结构研究已经取得的成果具有更多的相近成分 代数语义学和人工智能逻辑基础的研究广泛使用论域、赋值或指派的方法, 突出了计算的一面, 突出了逻辑意义上语言的语义模型的地位, 不仅实现了代数方法与逻辑方法在深层次的结合与统一, 而且, 有可能使得人工智能和智能计算研究未来使用的语言能够建立在严格的逻辑系统的语义模型之上 这是否预示着人工智能的逻辑基础研究目前主要集中在非经典的新型逻辑系统研究是一个正确的选择呢? 遗憾的是, 现有的各种逻辑系统要反映人的多种思维结构、计算和推理过程还远远不够

从另一个角度考虑, 尽管演化方法实际上在学科的发展中早就出现了, 其它学科也有类似的方法, 但是, 学术界在相当长的一个时期内没有注意到该方法的方法学意义 演化方法是一种特别适合计算科学研究与开发的技术方法 随着学科研究与发展的不断深化, 人们不可能在软硬件开发和应用中将所有的研究与开发工作都做得十分圆满, 万无一失 学科自身知识体系的复杂性决定了今后在研究与开发中需要不断地对先前进行的工作进行改进, 不断地完善, 这不仅仅是工程技术开发中一个具有普遍规律性的问题, 而且, 演化方法本身还可能具有重大的、酝酿着学术研究突破价值的科学意义 换言之, 针对软件开发的各个阶段发展的各种计算模型, 是否可以考虑根据计算环境的动态变化, 通过对模型的改造使其具有某种演化能力, 或者从一开始就考

虑设计具有演化能力的计算模型, 从初始状态出发, 根据软件开发各个阶段的不同要求, 逐步演化并支持各个不同阶段的要求 这样, 新计算模型就有可能变得具有演化计算能力, 表现出某些健壮性的特征和智能 要构造具有演化计算能力的计算模型, 恐怕突破传统计算模型的有穷状态集合或突破固定的运算操作有穷集合是关键 这是否暗示了智能计算系统的研究不必一定要沿着首先发展人工智能的逻辑基础, 然后发展智能计算机、智能计算系统的“演化”路线, 而存在着一条崭新的、以前从未有人涉足过的研究路线呢?

(2) 语言、程序与语义

研究计算, 离不开计算模型和语言 从 Chomsky 语言理论的角度看, 语言也是一种计算模型 通常, 人们从实现的角度考虑, 希望理论上的计算模型是一种具有状态转换特征的数学机器, 而从描述计算的角度考虑, 又希望计算模型是一种具有自然语言特征的高级语言 两者结合产生了高级程序设计语言及其编译理论与技术 由于计算的正确性常归结为计算程序语义的正确性问题, 因此, 用高级语言描述计算, 首先要确保描述程序的高级语言本身的语义是清楚的 可是, 语言的语义实际上是一个动态的概念, 每一个语句的语义与上下文和语境有关, 在不同的上下文和语境中句子的语义解释是不同的, 这就决定了静态翻译一个程序很难确保程序的语义完全正确, 特别是在分布式程序设计中 传统的单处理机计算机上的编译程序由于语境简单而容易处理程序的语义, 但具有多处理(器)机的并行与分布式计算机上的编译程序因不确定性带来的语境变化复杂而很难处理语言的语义 问题的关键是在语境动态变化的情况下不清楚一条语句的语义该作几种可行的解释中的何种解释, 这与日常生活中自然语言理解常常引起误解是类似的

在过去的几十年里, 已经发展了几种有代表性的形式语义学方法 对处理语义的方法进行分类, 存在内涵与外延等两种基本的方法, 但是, 如果从处理程序语义的过程分析, 除代数语义方法外基本上仍然是外延方法 内涵与外延是学科的典型方法, 处理程序的语义, 不太可能脱离上述基本的方法 现在的问题是, 一个程序的语义究竟是根据其内涵还是其外延来确定 目前的研究表明, 单纯考虑内涵或外延都存在局限性和不足 作者认为, 任何程序的语义, 无论是从内涵还是外延考察, 只有当两者保持一致时的语义才是程序正确的语义 这就是说, 从研究的

角度考虑,应该发展一种既能用内涵方式描述程序语言语义,又能用外延方式处理程序语言语义的计算模型。可惜这样一种模型至今尚未问世。

自然语言中的逻辑结构是语言学研究的学问,但是这个问题在高级程序设计语言及其语义的研究中未引起重视。事实上,今天所有高级语言的语义研究如果采用公理化方法,使用的逻辑系统均为经典的单调逻辑系统,这一点从高级语言循环结构成分的公理语义描述规则不难看出。虽然,非单调逻辑系统已经提出 20 多年了,但是迄今为止,从未有人基于非单调逻辑系统提出某种高级程序设计语言,也没有提出某种高级语言并为其建立非经典的逻辑基础(语义理论)。换言之,作为人工智能和新型计算机系统研究的一部分,应该超前研究基于非经典逻辑系统的高级语言、语法与语义,尽管这样一种可能存在的语言在表达上或许与人们的日常思维方式很不协调,但是,这种高级语言的研究将是重要的。

类似于这样的影响是很多的,限于篇幅,不再一一列举。

4.2 对学科人才培养的影响

学科的高速发展对大学专业人才的培养提出了挑战。学科持续、深入的高速发展,使得专业知识迅速积累,知识组织结构日益庞大,在内容上相对独立的专业课程科目总数已经达到 50 门左右,而且呈继续增长态势。面对急速变化中的学科和社会环境,学科专业教育与教学中过去沿用的外延发展模式已经不能适应学科发展的需要,专业教育与教学改革势在必行。针对计算科学,如何进行科学整理、分析与总结,采用内涵发展模式,系统研究并建立适应社会与学科协调发展需要的学科专业人才培养科学体系是一个亟待解决的问题^[5]。

学科教育与教学改革是一个十分复杂的系统,涉及到众多环节,涉及到对学科发展现状、特点和发展规律的整体把握,涉及到教育思想与人才培养目标、社会发展水平的影响,涉及到学科知识组织结构与教学规律的研究内容,涉及到管理体制与运行机制的科学性,而且,由于教育与教学改革研究成果最终要落实到人才培养的基本模式、人才培养的目标、教学计划与课程体系的改革方案上,受到专业学制年限、学生学习掌握学科内容的过程等时间、空间的制约,没有系统的、科学的方法,不可能使学科教育与教学质量产生质的飞跃。

在高等教育的办学实践中,从教育作为一项产业的角度观察,即使是西方发达国家,学科专业人才

的培养也还存在不少问题,许多方面仍处于探索之中,离建立一个学科人才培养的科学体系尚存在距离。西方发达国家长期沿用的自主办学、政府引导、社会评价、优胜劣汰的竞争机制,使得各研究大学完全根据自己的师资水平设置课程并开展教学。这样一种办学模式既有能使教学内容紧跟学科发展,倡导学术自由与争鸣的优点,也有与高等教育作为产业发展在规范性方面不相适应的缺点。对于高速发展的学科的本科教育,这种模式很难保证人才培养质量的稳定,总体上弊大于利。由于美国在学科方面处于领先地位,人们容易认同美国大学的学科专业教育。其实,美国在学科本科教育方面并没有多少先进的经验。一方面,由于大学生就业在美国是一个极其敏感的话题,大量一般用人单位的实用型、应用型倾向使得多数学校在学科专业人才培养方面比较偏向职业技术教育。另一方面,美国在学科研究与发展中获得学术人才的补充得益于其移民国家和多元文化。众所周知,美国每年从印度、中国、东南亚国家、欧洲、俄罗斯等一批国家网罗了大量学科发展后备人才和高级软硬件技术开发后备人才,美国大学本科学科人才培养工作上的不足不过是被学校大量科学技术研究的具体成果掩盖了而已。一旦美国失去了学科研究与发展所需要的学术后备人才的来源,其人才培养方案必将进行重大调整。因此,高等教育要实现既有规范与稳定的人才培养质量,又有学术自由与创新机制的科学技术产业,全面建立学科人才培养的科学体系将是教育与教学改革的必由之路。

一个学科人才培养的科学体系主要包括 3 方面的内容:学科人才培养的科学理论体系;基于学科人才培养的科学理论体系,由若干学校形成的分类分层次学科人才培养的示范基地;学科人才培养的质量保证体系。很明显,一旦建立了学科人才培养的科学体系,人才培养的质量将得到根本保证。

91 教学计划的相关研究报告中使用一个二维矩阵来定义和把握学科,即将学科的 3 个过程作为一维,学科的 9 个核心主科目作为另一维,指出 3 个过程在学科的各个层面上普遍出现,并建议用 12 个核心概念将学科的各个知识体串联起来,以达到对学科的整体把握与认识。2000 年,联合攻关小组在网上发表了一个 2001 方案(简称 2000 总体报告),增加了一些新的学科,将大量知识体(点)列出,供使用者选择。这样一种方法实际上是一种因素分析法,本质上仍然是一种外延的方法。随着学科新分支方

向的不断出现和知识的积累,面对大量的知识点,人们很难判定何为重要的,何为基本的。由于91教学计划相关研究报告并未解决内涵发展模式一些重要的、有争议的问题,如数学、实验在整个学科教学中的地位与作用问题,结果,ACM和IEEE/CS联合小组的理论研究成果与他们给出的学科分类科目结合在一起,给人以91教学计划和2000报告难于实际操作的印象和进一步研究的误导。

我们提出的学科教育与教学改革的指导思想是:从计算科学一级学科的知识组织结构出发,以国内外多年来学科教育和教学的经验和教训为线索,在认真分析学科专业教育与教学中存在的问题的基础上,用学科方法论的内容作为切入点,通过全面分析和总结学科发展的历程,整体上把握学科发展的现状、趋势、特点、内在规律及其对专业教育的影响,结合社会实际,明确不同层次、不同类别的人才培养目标,按照学科知识组织的层次结构和分类科目组织教学,将过去沿用的外延发展模式转变为内涵发展模式,重视基础,加强数学、物理学和计算机专业基础等核心知识的教学,精简课程体系,理论联系实际,强化实验教学,规范运作过程,兼顾正规教育与职业技术教育、高层次人才培养与社会各行各业对计算科学一般性应用人才的培养,实行分层次与分类办学。在德育教育方面,重视和加强学校文化建设,倡导科学理念,培养人文精神,创造优良的学风和文明的校风,主张环境熏陶、文化育人、自我管理、自我完善的方针。在校内管理体制和运行机制方面,依靠专家,突出学术,全面建立科学管理体系,确保学科教育与教学改革方案得到准确定位和平稳运行。在此基础上,从工程科学和教育产业的角度建立学科人才培养的质量保证体系,选择若干所学校进行学科专业教育与教学改革试点,按照教材建设理论研究成果的指导和新的运行机制,动员国内的力量,由中青年科学家和权威出版社创作、编辑、出版一套高起点的学科系列教材,逐步形成分层次分类的学科专业人才培养示范基地,全面建立学科人才培养的科学体系。

根据理论研究的成果和上述学科教育与教学改革的指导思想,培养高素质计算科学学科人才的重要环节主要包括下列几个方面:

(1) 打下坚实的数学基础,实现思维方式的数

学化;

(2) 系统掌握坚实的计算科学专业基础(技术)理论知识;

(3) 较好地掌握计算科学基本实验方法和技术;

(4) 培养正确的思想方法,注重全面发展
上述环节在实际操作中常归结为下列主要问题:

(1) 如何实现思维方式的数字化?
(2) 各学期重点课程包括哪些内容?
(3) 实验课程在学科专业教学计划中的地位和作用

(4) 如何提高专业能力?
(5) 理解科学与科学素养
有关这方面的详细内容,可阅读参考文献[2~5]

4.3 对学科教材建设的影响

学科方法论研究的进展必将对未来学科教材建设工作产生重大影响。今天,许多人已经意识到学科教材存在不少问题,普遍认为内容陈旧。但是,究竟教材存在的问题具体体现在哪些方面,内容陈旧的内涵是什么,众说纷纭,尚无定论。为此,一些出版社在新一代教材建设方面进行了积极的探索,出版了一些新的系列教材。

长期以来,国内外教材建设的一个不足之处是系列教材缺乏明确的指导思想和理论研究基础,不同的教材之间缺乏内容上避免重复的明确分工,缺乏内容上的前后呼应,缺乏风格上的一致和协调,也缺乏明确服务于某一人才培养体系和培养方案。

按照自顶向下设计,建立内涵发展优先,理论与实践并重,分层次分类办学的人才培养模式。培养方案,我们首先界定了当前研究生4门学位课程的科目和内容^[2~5]。4门学位课程是:高等逻辑、高等计算机体系结构、并行算法设计基础(或分布式算法设计基础)、形式语义学。由于4门学位课程的学时数均为72学时,其内容的学术深度对本科教学产生了很大的压力。例如,原先大批学校开设的离散数学(72~90学时)不可能同上述课程接轨。这就涉及到本科教材与内容需要进行系统的改革。现在面临的问题是:本科教材与研究生教材如何进行体现内涵发展模式的改革?

我们认为,欲体现内涵发展优先,理论与实践并重,分层次分类办学的内涵发展模式,教材的改革必须建立在深入的科学研究基础之上,必须建立在对

整个学科的认知基础之上,按照系统的教材建设指导思想和科学的运行机制,并辅之于相应的配套措施,才可能真正实现质的飞跃

教材中许多深入的研究是不可或缺的。由于学科高速发展,知识组织结构日益庞大,一旦按照内涵模式建设教材,势必要求作者保留最重要的也是最基本的内容。那么,哪些是学科教材必须保留的内容?哪些是应该删除的内容?哪些是可以作为学生自学扩展知识的内容?哪些又是可为各个学校自选而办出特色的内容?教材究竟如何体现学科方法论的内容?如何体现循序渐进和深入浅出?如何体现深度与广度的统一?这些问题将是建设一个优秀系列教材必须解决的问题。可惜,国内外学术界至今在这方面做出的努力还很不够。

致谢 作者衷心感谢教育部**陈祖福**、葛道凯同志,厦门大学教务处杨斌、林铁民、吕子玄、陈品健、洪艺敏等同志对 13-22 项目研究工作的支持,感谢 13-22 项目组各位教授对本项研究工作提出的大量中肯的、建设性的意见。

参 考 文 献

- Denning P J *et al*. Computing as a discipline. Communication of ACM, 1989, 32(1): 9~ 23
- 赵致琢. 高等学校计算科学教育. 北京: 科学出版社, 1996
(Zhao Zhizhuo. The Education of Computing Science in Universities(in Chinese). Beijing: Science Press, 1996)
- 赵致琢. 计算科学导论. 北京: 科学出版社, 1998
(Zhao Zhizhuo. Introduction to Computing Science (in Chinese). Beijing: Science Press, 1998)
- 赵致琢. 计算科学导论(第二版). 北京: 科学出版社, 2000
(Zhao Zhizhuo. The Education of Computer Science and Technology in University 2nd ed (in Chinese). Beijing: Science Press, 2000)
- 赵致琢. 高等学校计算机科学与技术学科专业教育(第二版). 北京: 科学出版社, 2000
(Zhao Zhizhuo. The Education of Computer Science and Technology in Universities (in Chinese). Beijing: Science Press, 2000)
- Gary L E. Profiling computer science master's programs 1991, 34(1): 100~ 109
- Turner A J *et al*. A summary of the ACM/IEEE-CS Joint Curriculum Task Force report: Computing curricula. Communication of ACM, 1991, 34(6): 68~ 84
- Cross II J H, Denning P J *et al*. Composition of the Curriculum 2001 Joint Task Force. Published in Electronic File on the Internet, 2000. 1~ 54
- Lee E A, Messerschmitt D G. Engineering and education for the future. IEEE Computer, 1998, (1): 77~ 85
- ACM. Curriculum'68: Recommendations for undergraduate program in computer science. Communication of ACM, 1968, 11(3): 151~ 197
- ACM. Curriculum'78: Recommendations for undergraduate program in computer science. Communication of ACM, 1979, 22(3): 147~ 166
- Cain J T *et al*. The IEEE computer society model program in computer science and engineering. Computer, 1984, 17(4): 8 ~ 17
- Engel G L. A comparison of the ACM/CS and the IEEE/CSE model curriculum subcommittee recommendations. Computer, 1977, 10(12): 121~ 123
- Magel K I *et al*. Recommendations for master's level programs in computer science. Communication of ACM, 1981, 24(3): 115~ 123
- Newell A *et al*. What is computer science. Science, 1967, 157: 1373~ 1374
- Nunamaker J E. Educational programs in information systems. Communication of ACM, 1981, 24(3): 124~ 133
- Nunamaker J E. Information system curriculum recommendations for the 80's: Undergraduate and graduate programs. Communication of ACM, 1982, 25(11): 781~ 805
- William F A. The development of computer science education. Advances in Computer, 1985, 24: 319~ 377
- 贝纳尔 J D. 科学的社会功能. 北京: 商务印书馆, 1986
(Bernal J D. The Social Function of Science (in Chinese). Beijing: Commercial Press, 1986)
- 李凯尔特 H. 文化科学和自然科学. 北京: 商务印书馆, 1991
(Von Rickert H. Kulturwissenschaft und Naturwissenschaft (in German). Beijing: Commercial Press, 1991)
- 王浩. 数理逻辑通俗讲话. 北京: 科学出版社, 1983
(Wang Hao. The Lectures on Mathematical Logic. Beijing: Science Press, 1983)
- 威廉 N, 玛莎 N. 逻辑学的发展. 北京: 商务印书馆, 1995
(William K, Martha K. The Development of Logic (in Chinese). Beijing: Commercial Press, 1995)
- 黄铠, 徐志伟. 可扩展的并行计算技术、结构与编程. 北京: 机械工业出版社, 2000
(Huang Kai, Xu Zhiwei. Scalable Parallel Computing Technology, Architecture, Programming (in Chinese). Beijing: Chinese Machine Press, 2000)
- Michael J Q. Designing Efficient Algorithms for Parallel Computers. New York: McGraw-Hill, 1987
- Lynch N A. Distributed Algorithms. Morgan Kaufmann, 1996

- 26 Hoare C A R. Communicating Sequential Processes. Englewood Cliffs, NJ: Prentice-Hall, 1985
- 27 Gerard T. Introduction to Distributed Algorithms. Cambridge: Cambridge University Press, 1994
- 28 Barendregt H P. The Lambda Calculus: Its Syntax and Semantics, 2nd ed. Amsterdam: North-Holland, 1984
- 29 Huang K. Advanced Computer Architecture: Parallelism, Scalability, and Programmability. New York: McGraw-Hill, 1993

- 30 陆汝钊. 计算机语言的形式语义. 北京: 科学出版社, 1992
(Lu Ruqian. The Formal Semantics of Computer Programming Languages(in Chinese). Beijing: Science Press, 1992)



赵致琢 男, 1957 年生, 博士, 主要从事计算模型、分布式计算与程序理论、计算机教育方面的研究

2001 全国开放式分布与并行计算学术会议 征文通知

本届学术会议(简称DPCS2001)由中国计算机学会开放系统专委会主办、南京大学计算机科学与技术系承办,定于2001年10月24日至26日在南京召开。

征文范围

会议征文范围涉及开放系统的广泛领域,重点是开放式分布与并行计算的理论、实践与应用,主要包括:

- (1) 开放式分布与并行计算系统,包括这类系统的体系结构、算法与优化、语言与编译、性能模拟与评价、操作系统以及系统中的存储器层次结构、系统平台的程序设计环境和系统的应用等;
- (2) 多媒体的开放式并行与分布处理技术;
- (3) 开放式网络技术与应用,包括开放式网络计算、统一网络、Internet 等技术与应用;
- (4) 上述重点领域有关科学技术与应用的发展概况与方向

文稿要求

会议只录取尚未正式发表过的论文。论文一律沿用期刊《计算机研究与发展》的格式,一般不超过5000字,一律为A4激光打印稿并随寄软盘片(word格式)。论文一式2份,请自留原稿。论文须于2001年6月1日前寄达陈贵海收,务必在信封左下角注明DPCS2001。

联系方式

- (1) 南京地区联系人: 陈贵海
联系地址: 南京大学计算机科学与技术系
联系电话: (025) 3592339, 3595652(家), 13705151348(手机)
邮政编码: 210093
传 真: (025) 3300710
Email: gchen@nju.edu.cn(务必在邮件主题中注明DPCS2001)
会议主页: <http://cs.nju.edu.cn/~gchen/dpcs2001.html>
- (2) 北京地区联系人: 陈炳从(开放系统专委会主任)
联系地址: 北京619信箱63号
联系电话: (010) 62311951
邮政编码: 100083

中国计算机学会开放系统专委会
2000. 7. 5